

## 並列言語からみた並列計算機の通信アーキテクチャ

2 G-6

岩本善行 大栗和久 吉永努 馬場敬信

宇都宮大学工学部

## 1 はじめに

これまでにさまざまな並列計算機のアーキテクチャが提案され、それぞれのアーキテクチャに基づいて並列言語および並列処理を実現するためのライブラリが独自に開発されてきた。現在、MPIなどの通信ライブラリが標準化され、実装されているが、個々の並列計算機のアーキテクチャとの適切な対応が取れていないため、それぞれの並列計算機に実装した場合にはその性能を十分に引き出しているとは言い難く、このためにさまざまな研究が行われている[1]。

本研究では、並列計算機アーキテクチャとその上に実装されている独自の並列化ライブラリを分析し、高性能な並列計算機アーキテクチャや並列言語、並列化ライブラリを提案することを目的としている。このための第一段階として商用並列計算機富士通AP1000[2]およびNEC Cenju-3[3]を取り上げ、初めに、その計算機独自の言語ライブラリCell OSライブラリとPARALIB/CJの通信制御命令の基本的な性能を得る。さらに、並列にデータを処理する場合に必要な基本的な機能を、実際にそれぞれの機種上でそれぞれ独自のライブラリを用い、アルゴリズムを変えて実現し、その通信性能を得る。最後に各ライブラリでよく使用される命令を取り上げ、その実現方法をユーザレベルのプログラムから、OSレベル、カーネルレベルまで階層的に分析する。これらの評価によって、それぞれの機種のノード間接続網の違いなどの通信アーキテクチャと、通信ライブラリの命令が適切に対応しているか、また、性能のネックとなっているのはどの部分なのかなどを分析する。

## 2 AP1000およびCenju-3の概要

## 2.1 通信ライブラリ

AP1000およびCenju-3はそれぞれ独自に開発された通信ライブラリを持ち、AP1000ではCell OSライブラリ、Cenju-3ではPARALIB/CJと呼ばれる。それぞれの機械においてユーザに提供されているライブラリに特徴的なC言語における関数を表1に示す。

## 2.2 ノード間接続網

AP1000におけるノード間接続網を図1に示す。AP1000には各ノードをトラス状に接続するT-NET、全

表1 主な通信制御命令

|       | AP-1000            | Cenju-3               |
|-------|--------------------|-----------------------|
| 1対1通信 | l_asend<br>l_arecv | Cjrmwrite<br>Cjrmread |
| 1対多通信 | cbroad<br>xy_isum  | Cjrmwritem            |
| 同期    | sync               | Cjbarrier             |
| タスク生成 |                    | Cjrpc                 |

セルに対してブロードキャストを行うB-NETとS-NETが用意されている。

Cenju-3では4×4のクロスバスイッチを使用した

多段接続網を用いている(図2)。この接続によって全ノード間の距離は等しくなる。現在はクロスバスイッチのブロードキャスト機構を使用していないため、全ノードに対するブロードキャストはOSレベルで逐次的に行なう必要がある。

## 3 通信性能評価

上記2機種上において、並列計算において必要となる以下の機能についてプログラムを作成し、その実行時間を測定した。

## (1) メッセージ送受信

メッセージ送受信処理性能として次の4種類について測定した。測定には内部時計を使用し、送信処理開始から、メッセージが到着して使用可能となるまでの時間である。

## 1. 2ノード間メッセージピンポン (図3)

## 2. Shift

全ノードが隣接ノードへデータを送信する

## 3. Complete exchange

全ノードが、全ノードに対し配列に用意した異なったデータを送信する。

## (2) ブロードキャスト (図4)

各機種に用意されているライブラリ関数を使用し、0番のノードから全ノードに対してメッセージを送信する。

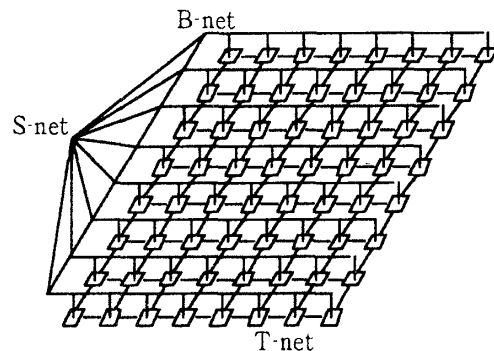


図1 AP-1000におけるノード間接続網

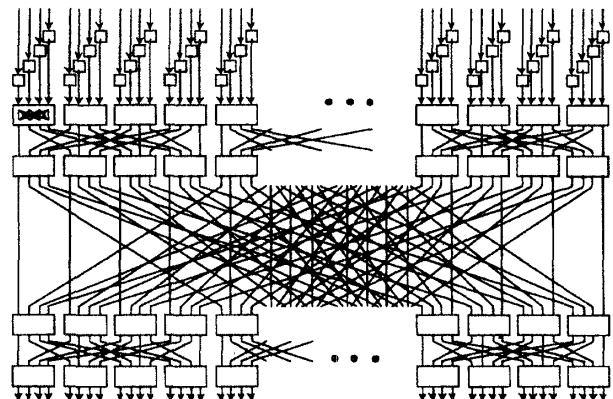


図2 Cenju-3におけるノード間接続網

### (3) バリア同期 (図5)

ノード数を変化させ、10000回の同期に必要な時間を計測し、その1回当たりの実行時間を求める。

### (4) リダクション (図6)

全ノードに用意したdouble型のデータの総和を求める。1つのノードへ集めて演算後、全ノードに送り出すMaster-Slave方式 (図中のMS)と、ネットワークを段階的に用いた演算アルゴリズムCascade方式とShuffle Exchange方式 (SE)、AP1000に用意されているxy\_dsumを用いた場合の4通りについて1回当たりの実行時間を計測する。

## 4 考察

上記の結果から、1対1のデータ送受信においては(図3)、メッセージサイズが小さい場合にはAP1000が有利であるが、サイズによる時間の増加はCenju-3が低い。これは、ユーザレベルで相手の受信アドレスを決定するCenjuに対して、AP1000ではメッセージを受信後、さらに受信処理をユーザが指定し、その格納先に対してさらにコピーしなければならないことが挙げられる。すなわち、ユーザレベルで受信相手先のアドレスを決定するようなライブラリを用意することによって高速化が期待できる。さらに、Cenju-3のOSレベルではメッセージのサイズをキャッシュの容量やDMA転送の使用と関連させて5通りに場合分けを行ない高速化を試みている。このことが図3におけるグラフの200バイト以下で転送時間が増加している原因と考えられる。

次に1対多の場合 (図4、図5)、S-netを用意しているAP1000が有利になり、同様にバリア同期においてもAP1000はノード数によらず一定となる。

リダクション処理では (図6)、AP1000に用意されているライブラリが最も高速である。ただし、ライブラリとして用意されていない場合には、多段接続網では全ノードから1つのノードにデータを集め、演算後送り返すようなMaster-Slave型が有利になるが、トーラス網で演算を行なうAP1000では、そのネットワークをフルに活用したCascadeまたはShuffle Exchange型が有利となる。これは、Cenju-3ではネットワーク内でメッセージが衝突する可能性があるが、AP1000では隣接ノードに対してはすべてのノードに対して接続が確保されていることによるものと思われる。

## 5 おわりに

本稿では商用並列計算機として富士通AP1000およびNEC Cenju-3のメッセージ転送性能をユーザレベルから定量的に評価し、さらにOS内の処理の観点からそれぞれの特性について検討した。データ転送先アドレスの直接指定、メッセージサイズによる処理の細分化、ノード間接続網とリダクション処理アルゴリズムの対応などが、ユーザレベルから見た通信性能の高速化に有効であることの見通しを得た。今後は、転送データのメモリ配置の変化などによる通信条件をさまざまに変化させたときの実行時間の測定や、CJrpcなどの機種に依存したライブラリ機能の評価などを行ない、これらをもとに高速化のための具体的な方法を検討する予定である。

## 謝辞

本研究は、一部文部省科学研究費、基盤(C)08680346、電気通信普及財団、並列・分散処理研究推進機構の援助による。

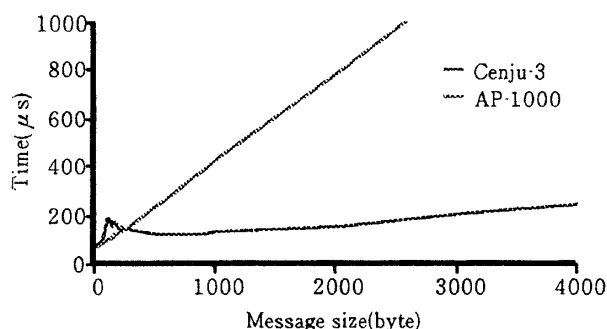


図3 2node pingpongにおける通信時間

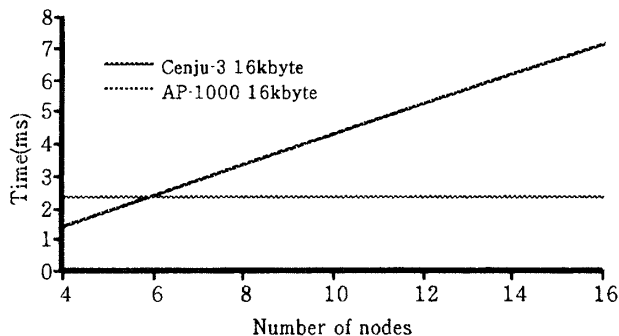


図4 broadcastにおけるノード数と実行時間

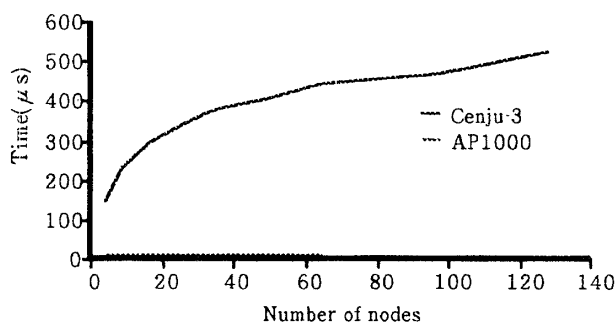


図5 バリア同期のノード数と実行時間

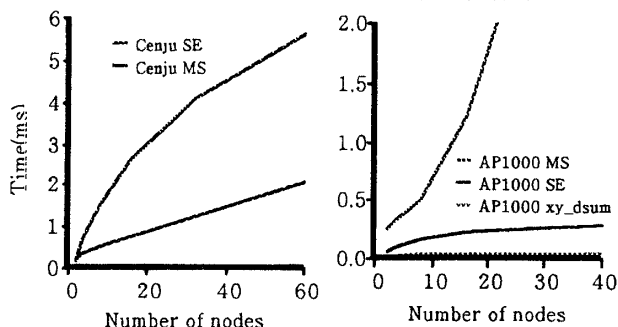


図6 リダクション処理のノード数と実行時間の結果

## 参考文献

- [1] 田中好夫、久保田和人、佐藤三久、関口智嗣: 並列アルゴリズムにおけるCollective通信の性能比較, SWoPP'96 HPC62-6, pp.19-pp.26(1996)
- [2] David Sitsky, Kenichi Hayashi: Implementing MPI for the Fujitsu AP1000/AP1000+ using Polling, Interrupts and Remote Copying, JPSS '96 pp.177-pp.184(1996)
- [3] 広瀬哲也、細見岳生、丸山勉、加納健: 並列コンピュータCenju-3のプロセッサ間通信方式とその評価, JSPP'95 pp.241-pp.248(1995)